Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

Claim 1 (canceled).

2 (Previously presented): The method of Claim 12 wherein:

said fixed-point operand is one of at least two fixed-point operands to be used by said instruction; and

said expanding comprises normalization of at least said fixed-point operand if said fixed-point operand has said property of a value different from another value of said property of another operand to be used by said instruction.

3(Previously presented): The method of Claim 12 wherein:

said instruction is to use two operands, with

said fixed-point operand as a first operand, and another fixed-point operand as a second operand; and

said first operand has a property of a first value and the second operand has said property of a second value different from said first value, said expanding comprises normalization of at least one fixed-point operand to have a common value for said property, said common value being one of the first value and the second value.

4(Previously presented): The method of Claim 12 wherein:

the fixed-point result has said value of at least said one property.

5(Previously presented): The method of Claim 12 further comprising:

said computer determining a property value for the fixed-point result, based on said at least one property value of the fixed-point operand, said fixed-point result being displayed to said user based on said property value from said determining.

6(Previously presented): The method of Claim 12 comprising:

said computer determining a property value for the corresponding fixed-point result, based on the instruction that was performed on the fixed-point operand.

7(Previously presented): The method of Claim 12 wherein:

the fixed-point representation includes the value of the fixed-point number in memory in floating-point representation.

8(Previously presented): The method of Claim 12 wherein:

the fixed-point representation includes a value in memory to represent the signedness property.

9(Previously presented): The method of Claim 12 wherein:

the fixed-point representation includes a value in memory to represent the complexness property.

10(Previously presented): The method of Claim 12 wherein:

the fixed-point representation uses at least two locations in memory to store a precision of the value.

11(Previously presented): The method of Claim 12 wherein:

the fixed-point representation includes a value in memory to represent a scaling factor for the fixed-point number.

12 (Previously presented): In a computer comprising a memory and a floating point unit, a method implemented in a set of instructions executable by said computer, the method comprising:

said computer receiving a program, a portion of said program comprising an operand (hereinafter "fixed-point operand") represented in a fixed-point representation;

wherein said fixed-point operand has at least one property selected from a group consisting of (signedness, precision, complexness);

said computer expanding said fixed-point operand into a floating-point representation to obtain a floating-point equivalent;

said computer storing a precision of the fixed-point operand;

said computer further receiving in said portion an instruction comprising an operation to be performed on the fixed-point operand by a fixed point processor;

said computer using said floating point unit to perform on the floating-point equivalent, at least one floating-point operation that corresponds to the fixed-point operation, yielding at least one floating-point result;

said computer reducing said at least one floating-point result generated by the floating-point operation into a fixed-point result;

said computer using the stored precision during reduction of the floating-point result into the fixed-point result; and

said computer displaying said fixed-point result to a user to enable said portion of said program to be debugged without use of said fixed point processor.

13(Previously presented): The method of Claim 12 wherein during reduction of the floating-point result to the fixed-point result, the method comprises:

said computer using a predetermined storage element to identify a mode of rounding to be performed on the floating-point result, wherein the mode of rounding is one of: round (round-to-nearest), fix (round towards zero), ceil (round towards positive infinity), and floor (round towards negative infinity), and said computer performing said rounding, and said fixed-point result being displayed after said rounding is performed.

14(Previously presented): The method of Claim 12 wherein during reduction of the floating-point result into the corresponding fixed-point result, the method comprises:

said computer using a predetermined storage element to identify a kind of arithmetic to be performed on the floating-point result, wherein the kind of arithmetic is one of: saturation and modulo.

15(Previously presented): The method of Claim 12 wherein:

said fixed-point representation is hereinafter "first fixed-point representation";

the fixed-point result is expressed in a second fixed-point representation which is different from the first fixed-point representation; and

the method further comprises said computer using a predetermined storage element in said memory to identify a property of the second fixed-point representation, said fixed-point result being displayed to said user in said second fixed-point representation.

16(Previously presented): The method of Claim 15 wherein:

said property of the second fixed-point representation is precision.

17(Previously presented): The method of Claim 15 wherein:

said property of the second fixed-point representation is signedness.

18(Previously presented): The method of Claim 15 wherein:

said property of the second fixed-point representation is complexness.

19(Previously presented): The method of Claim 12 wherein during expansion of the fixed-point operands into floating-point equivalents, the method comprises:

said computer detecting that the operands are invalidly scaled and issuing a warning message displayed to said user based on a predetermined storage element.

20(Previously presented): The method of Claim 12 further comprising:

said computer using at least the precision of the fixed-point operand, during emulation of another instruction that uses a result of the fixed-point arithmetic operation.

21 (Previously presented): The method of Claim 12 wherein:

the floating-point representation conforms to an IEEE Standard for floating-point arithmetic.

22 (Previously presented): The method of Claim 12 wherein the instruction is to be performed on said fixed-point operand and at least an additional floating-point operand, and the fixed-point arithmetic operation is to be performed on the fixed-point operand and said additional floating-point operand, and the method further comprises:

during the act of receiving said floating-point operand, said computer reducing said additional floating-point operand into fixed-point representation, based on the precision of the fixed-point operand.

23 (Previously presented): The method of Claim 22 wherein:

reduction of the floating-point operand into said fixed-point representation is based on said property of the fixed-point operand.

24(Original): The method of Claim 23 wherein:

said property is precision.

25 (Original): The method of Claim 23 wherein:

said property is signedness.

26 (Original): The method of Claim 23 wherein:

said property is complexness.

27 (Previously presented): A method implemented in a set of instructions executable by a computer that comprises a floating point unit, the method comprising:

said computer receiving receiving a program, at least a portion of said program being written for execution on a fixed point processor, said fixed point processor being not available for execution of said program, said portion of said program comprising at least one operand represented in fixed-point representation (hereinafter "fixed-point operand"), said fixed-point operand having at least one property selected from a group consisting of (signedness, precision, complexness);

said computer expanding said fixed-point operand into a floating-point representation to obtain a floating-point equivalent;

said computer receiving an instruction comprising an operation to be performed on the fixed-point operand;

said computer receiving another instruction that indicates a type of said fixed-point operand;

wherein said another instruction comprises a call to a function;

said computer using said floating point unit to perform on the floating-point equivalent, at least one floating-point operation that corresponds to the fixed-point operation, yielding at least one floating-point result;

said computer reducing, based on a value of said at least one property, said at least one floating-point result generated by the floating-point operation into a fixed-point result; and

said computer displaying the fixed point result and receiving instructions for debugging said portion of said program without use of the fixed point processor.

28 (Original): The method of Claim 27 wherein:

said function comprises instantiation of an object of a predetermined class, the object comprising said floating-point equivalent and at least one property of said fixed-point operand.

29 (Previously presented): The method of Claim 27 wherein:

the fixed-point operand is a real number;

the method further comprises said computer receiving another indication via another function name that a complex number is to be expressed in fixed-point representation; and

on receipt of an imaginary part and a real part of the complex number, said computer expanding each part into a corresponding floating-point equivalent.

30 (Previously presented): A method implemented in a set of instructions executable by a computer that comprises a floating point unit, the method comprising:

said computer receiving receiving a program, at least a portion of said program being written for execution on a fixed point processor, said fixed point processor being not available for execution of said program, said portion of said program comprising at least one operand represented in fixed-point representation (hereinafter "fixed-point operand"), said fixed-point representation having a precision identifying a first plurality of bits on a left side of a point and a second plurality of bits on a right side of said point;

said computer expanding said fixed-point operand into a floating-point representation to obtain a floating-point equivalent;

said computer receiving in said program an instruction comprising an operation to be performed on the fixed-point operand;

wherein said instruction comprises overloading of an operator normally used to denote said corresponding floating-point operation;

said computer using said floating point unit to perform on the floating-point equivalent, at least one floating-point operation that corresponds to the fixed-point operation, yielding at least one floating-point result;

said computer reducing, based said precision, said at least one floating-point result generated by the floating-point operation into a fixed-point result; and

said computer displaying the fixed point result and receiving instructions for debugging said portion of said program without use of the fixed point processor.

Claims 31-39 (canceled).

40 (Previously presented): In a computer comprising a floating point unit, a method implemented in a set of instructions executable by said computer, the method comprising:

said computer receiving at least one operand represented in fixed-point representation (hereinafter "fixed-point operand");

said computer expanding said fixed-point operand into a floating-point representation to obtain a floating-point equivalent;

said computer receiving an instruction comprising a fixed-point operation to be performed on the fixed-point operand by a fixed point processor;

said computer using said floating point unit to perform on the floating-point equivalent, at least one floating-point operation that corresponds to the fixed-point operation, yielding at least one floating-point result;

said computer reducing said at least one floating-point result generated by the floating-point operation into a fixed-point result; and

said computer displaying said fixed-point result and receiving instructions to debug a portion of a program comprising said fixed-point operand and said fixed-point operation without use of said fixed point processor;

wherein if the fixed-point operation is specified as saturation arithmetic and if said floating-point result is a vector or array, said act of reducing comprises:

said computer setting a maximum value and a minimum value for a fixed-point result to be obtained from reducing the floating-point result generated by the floating-point operation, based on the precision and signedness of at least said fixed-point operand;

said computer replacing any negative numbers in the floating-point result with zero;

said computer replacing any numbers in the floating-point result that are greater than the maximum value with the maximum value;

said computer replacing any numbers in the floating-point result that are less than the minimum value with the minimum value.

41 (Original): The method of Claim 40 wherein if a rounding mode is specified as round, the act of reducing further comprises:

subsequent to performance of said act of setting, rounding the floating-point result.

42 (Previously presented): A computer-implemented method implemented in a set of instructions, the computer-implemented method comprising:

receiving a program, at least a portion of said program being written for execution on a fixed point processor, said fixed point processor being not available for execution of said program, said portion of said program comprising at least one operand representing a real number expressed in fixed-point representation (hereinafter "fixed-point operand"), said fixed-point representation having a precision identifying a first plurality of bits on a left side of a point and a second plurality of bits on a right side of said point;

said computer expanding said fixed-point operand into a floating-point representation to obtain a floating-point equivalent, said expanding comprising storing in said memory said precision of said fixed-point operand;

said computer further receiving in said portion of said program an instruction to be performed on the fixed-point operand;

said computer using said floating point unit to perform on the floating-point equivalent, at least one floating-point operation that corresponds to the fixed-point operation, yielding at least one floating-point result;

said computer reducing, based on kind of said instruction received, said at least one floating-point result generated by the floating-point operation into a fixed-point result; and

said computer displaying said fixed-point result to a programmer to enable said portion of said program to be debugged without use of said fixed point processor.

43 (Original): The method of Claim 42 wherein:

said instruction includes an operator and said act of reducing is based on said operator.

44 (Currently amended): A computer readable memory encoded with a plurality of objects, each object representing at least one fixed-point number, each object being encoded in a plurality of locations comprising:

a first location encoded with a value of a signedness property of said fixed-point number;

a second location encoded with a value of a complexness property of said fixed-point number;

a plurality of locations encoded with values of subproperties of a precision property of said fixed-point number; and

at least one location encoded with a floating-point value of said fixed-point number;

wherein the memory further comprises instructions that cause a computer to operate floating point hardware to use said floating-point value

and display a result of said use thereby to help in debugging at least a portion of a program without use of a fixed point processor.

45 (Original): The memory of Claim 44 wherein each object further comprises:

a third value of a scaling factor of said fixed-point number.

46 (Previously presented): The memory of Claim 44 wherein at least one object in said plurality further comprises:

a plurality of additional floating-point values;

wherein said values of said properties are identical for each of said additional floating-point values, and said at least one object represents a vector operand.

47 (Original): The memory of Claim 44 wherein said precision property comprises:

a number of bits to the left of a point in the fixed-point number as a subproperty.

48 (Previously presented): A memory encoded with a plurality of objects, each object representing at least one fixed-point number, each object being encoded in a plurality of locations comprising:

a first location encoded with a value of a signedness property of said fixed-point number;

a second location encoded with a value of a complexness property of said fixed-point number;

a plurality of locations encoded with values of subproperties of a precision property of said fixed-point number; and

at least one location encoded with a floating-point value of said fixed-point number;

wherein said precision property further comprises a number of bits to the right of said point in the fixed-point number as another subproperty;

wherein the memory further comprises a program written for execution on a fixed point processor, said fixed point processor being not available for execution of said program, said program comprising instructions to:

use floating-point hardware on the floating-point value in said at least one location to generate a floating-point result; and

display said fixed-point result to a user by use of at least said precision property to enable at least a portion of said program to be debugged without use of said fixed point processor.

49 (Original): The memory of Claim 47 wherein said precision property further comprises:

a total number of bits in the fixed-point number as another subproperty.

50 (Previously presented): The memory of Claim 44 wherein said precision property further comprises:

a number of bits to the right of a point in the fixed-point number as a subproperty.

51 (Original): The memory of Claim 44 wherein said at least one location encoded with said floating-point value of said fixed-point number holds a real component of said fixed-point number, the memory further comprising:

at least one additional memory location for holding a complex component of said fixed-point number.

52 (Previously presented): A computer that supports floating-point arithmetic, the computer comprising:

means (hereinafter "receiving means") for receiving and storing in a memory of said computer a program written for execution on a fixed point processor, said fixed point processor being not available for execution of said program, said program comprising at least one operand represented in fixed-point representation (hereinafter "fixed-point operand"), said fixed-point operand having at least one property selected from a group consisting of (signedness, precision, complexness);

means, coupled to said receiving means, for expanding said fixed-point operand into a floating-point representation to obtain a floating-point equivalent;

said program in said memory comprising a fixed-point operation to be performed on the fixed-point operand;

a floating point unit coupled to the means for expanding to receive therefrom the floating-point equivalent and to generate at least one floating-point result by performance of at least one floating-point operation that corresponds to the fixed-point operation;

means for reducing said at least one floating-point result generated by the floating-point operation into a fixed-point result; and

means for displaying the fixed-point result and receiving instructions for debugging said program without use of the fixed point processor;

wherein said memory is coupled to each of said means and said floating point unit, said memory being encoded with a plurality of objects, at least one object representing at least said fixed-point operand, said object being encoded in a plurality of locations comprising:

a first location encoded with a value of a signedness property of said fixed-point operand;

a second location encoded with a value of a complexness property of said fixed-point operand;

a group of locations encoded with values of subproperties of a precision property of said fixed-point operand; and

at least one location encoded with a floating-point value of said fixed-point operand.

53 (Previously presented): In a computer, a method implemented in a set of instructions executable by said computer, the method comprising:

said computer receiving a program, at least a portion of said program being written for execution on a fixed point processor, said fixed point processor being not available for execution of said program, said portion of said program comprising at least one operand representing a real number expressed in fixed-point representation (hereinafter "fixed-point operand"), said fixed-point representation having a precision

identifying a plurality of first bits of the fixed-point operand on a first side of a point and a plurality of second bits of the fixed-point operand on a second side of the point;

said computer adding to said program a sequence of instructions to expand said fixed-point operand into a floating-point representation to obtain a floating-point equivalent, said sequence comprising an instruction to store in said memory said precision of said fixed-point operand;

said computer adding to said program at least a first instruction to use a floating point unit to perform on the floating-point equivalent, at least one floating-point operation that corresponds to the fixed-point operation, to yield at least one floating-point result;

said computer adding to said program at least a second instruction to reduce, based on said precision, said at least one floating-point result generated by the floating-point operation into a fixed-point result;

and

said computer displaying said fixed-point result and receiving instructions for debugging said program without use of said fixed point processor.

54 (Previously presented): The method of Claim 12 wherein said fixed-point operand comprises:

a value of a scaling factor of said fixed-point number.

55 (Previously presented): The method of Claim 12 wherein:

said fixed-point operand comprises a plurality of floating-point values; and

said fixed-point operand is a vector operand.

56 (Previously presented): The method of Claim 12 wherein said property comprises at least two of:

a number of bits to the left of a point in the fixed-point number;

a number of bits to the right of said point in the fixed-point number; and

a total number of bits in the fixed-point number.

57 (Previously presented): The method of Claim 12 wherein at least one location in said memory of said computer encoded with said floating-point value of said fixed-point number holds a real component of said fixed-point number, said memory further comprising:

at least one additional location for holding a complex component of said fixed-point number.

58 (Previously presented): The method of Claim 15 wherein said fixed-point operand comprises:

a value of a scaling factor of said fixed-point number.

59 (Previously presented): The method of Claim 15 wherein:

said fixed-point operand comprises a plurality of additional floating-point values; and

said fixed-point operand is a vector operand.

60 (Previously presented): The method of Claim 15 wherein said property of the second fixed-point representation comprises at least two of:

a number of bits to the left of a point in the fixed-point number;

a number of bits to the right of said point in the fixed-point number; and

a total number of bits in the fixed-point number.

61 (Previously presented): The method of Claim 15 wherein at least one location in said memory of said computer encoded with said floating-point value of said fixed-point number holds a real component of said fixed-point number, said memory further comprising:

at least one additional location for holding a complex component of said fixed-point number.

62 (Previously presented): The method of Claim 23 wherein said fixed-point operand comprises:

a value of a scaling factor of said fixed-point number.

63 (Previously presented): The method of Claim 23 wherein said fixed-point operand comprises:

a plurality of floating-point values; and

said fixed-point operand is a vector operand.

64 (Previously presented): The method of Claim 23 wherein said property comprises at least two of:

a number of bits to the left of a point in the fixed-point number;

a number of bits to the right of said point in the fixed-point number; and

a total number of bits in the fixed-point number.

65 (Previously presented): The method of Claim 23 wherein at least one location in said memory of said computer encoded with said floating-point value of said fixed-point number holds a real component of said fixed-point number, said memory further comprising:

at least one additional location for holding a complex component of said fixed-point number.

66 (Previously presented): The method of Claim 28 wherein each fixed-point operand comprises:

a value of a scaling factor of said fixed-point number.

67 (Previously presented): The method of Claim 28 wherein:

said fixed-point operand further comprises a plurality of additional floating-point values; and

said values of said properties are identical for each of said additional floating-point values, and said fixed-point operand is a vector operand.

68 (Previously presented): The method of Claim 28 wherein said property comprises at least two of:

a number of bits to the left of a point in the fixed-point number;

a number of bits to the right of said point in the fixed-point number; and

a total number of bits in the fixed-point number.

69 (Previously presented): The method of Claim 28 wherein at least one location in a memory of said computer is encoded with said floating-point value of said fixed-point number and holds a real component of said fixed-point number, the memory further comprising:

at least one additional memory location for holding a complex component of said fixed-point number.

70 (Previously presented): The method of Claim 30 wherein said fixed-point operand comprises:

a value of a scaling factor of said fixed-point number.

71 (Previously presented): The method of Claim 30 wherein:

said fixed-point operand comprises a plurality of floating-point values; and

said fixed-point operand is a vector operand.

72 (Previously presented): The method of Claim 30 wherein said property comprises at least two of:

a number of bits to the left of a point in the fixed-point number;

a number of bits to the right of said point in the fixed-point number; and

a total number of bits in the fixed-point number.

73 (Previously presented): The method of Claim 30 wherein at least one location in said memory of said computer encoded with said floating-point value of said fixed-point number holds a real component of said fixed-point number, said memory further comprising:

at least one additional location for holding a complex component of said fixed-point number.

74 (Previously presented): The method of Claim 40 wherein said fixed-point operand comprises:

a value of a scaling factor of said fixed-point number.

75 (Previously presented): The method of Claim 40 wherein:

said fixed-point operand comprises a plurality of floating-point values; and

said fixed-point operand is a vector operand.

76 (Previously presented): The method of Claim 40 wherein said property comprises at least two of:

a number of bits to the left of a point in the fixed-point number;

a number of bits to the right of said point in the fixed-point number; and

a total number of bits in the fixed-point number.

77 (Previously presented): The method of Claim 40 wherein at least one location in said memory of said computer encoded with said floating-point value of said fixed-point number holds a real component of said fixed-point number, said memory further comprising:

at least one additional location for holding a complex component of said fixed-point number.

78 (Previously presented): The memory of Claim 48 wherein each object further comprises:

a third value of a scaling factor of said fixed-point number.

79 (Previously presented): The memory of Claim 48 wherein at least one object in said plurality further comprises:

a plurality of additional floating-point values;

wherein said values of said properties are identical for each of said additional floating-point values, and said at least one object represents a vector operand.

80 (Previously presented): The memory of Claim 48 wherein said precision property comprises at least two of:

a number of bits to the left of a point in the fixed-point number;

a number of bits to the right of said point in the fixed-point number; and

a total number of bits in the fixed-point number.

81 (Previously presented): The computer of Claim 52 wherein:

a location in said plurality of locations is encoded with a value of a scaling factor.

82 (Previously presented): The computer of Claim 52 wherein:

said fixed-point operand comprises a plurality of floating-point values; and

said fixed-point operand is a vector operand.

83 (Previously presented): The computer of Claim 52 wherein said property comprises at least two of:

a number of bits to the left of a point in the fixed-point number;

a number of bits to the right of said point in the fixed-point number; and

a total number of bits in the fixed-point number.

84 (New): The memory of Claim 44 wherein:

a value of a variable in said portion of said program is displayed by said instructions.

85 (New): The memory of Claim 44 wherein:

a property of a variable in said portion of said program is displayed by said instructions.

86 (New): The memory of Claim 44 wherein:

said instructions evaluate an operation on a variable in said portion of said program and said result is from evaluation of said operation.

87 (New): The memory of Claim 86 wherein:

a property of said result is displayed by said instructions.

88 (New): The memory of Claim 44 wherein:

said instructions are implemented in an interpreter.

89 (New): The memory of Claim 44 wherein:

a warning is displayed by said instructions.

90 (New): The method of Claim 12 wherein:

said computer displays during said displaying, a value of a variable in said portion of said program.

91 (New): The method of Claim 12 wherein:

said computer displays during said displaying, a property of a variable in said portion of said program.

92 (New): The method of Claim 12 wherein:

said computer evaluates an operation on a variable in said portion of said program and during said displaying displays a result of evaluation of said operation.

93 (New): The memory of Claim 12 wherein:

said computer evaluates an operation on a variable in said portion of said program and during said displaying displays a property of a result of evaluation of said operation.

94 (New): The memory of Claim 12 wherein:

said set of instructions are implemented in an interpreter.

95 (New): The memory of Claim 12 wherein:

a warning is displayed by said set of instructions.